## Xpath定位方法

* 通过元素类型和属性定位：//<element>[ @<attribute>=”<value>” ]；

### 模糊定位

* Contains方法：//<element>[ contains( @<attribute>, ”<value>” ) ]；
* Start-with方法：//<element>[ start-with( @<attribute>, “<value>” ) ]；
* Text方法：//<element>[ text()=”value” ] || //<element>[ contains(text(), “<value>”) ]；

## Docker

### 容器

* 启动容器：docker run (-d,-it,--name) <image> (d后台模式,it交互模式,name命名)；
* 退出容器：exit/Ctrl + D；
* 获取镜像：docker pull <image>；
* 查看所有容器：docker ps -a；
* 停止一个容器：docker stop <container ID>；
* 启动一个已停止的容器：docker start <container ID>；
* 重启容器：docker restart <container ID>；
* 进入一个后台容器：docker attach；
* 在一个正在运行的容器中执行命令：docker exec <container name> <command>；
* 删除容器：docker rm -f <container ID>;
* 清除所有已停止的容器：docker container prune；
* 启动容器时映射端口（随机分配）：docker run -P <app’s port>；
* 启动容器时映射端口（指定地址/端口/协议）：docker run -P<host’s IP>: <app’s port>:<host’s port>/<udp>；
* 查看容器的网络端口：docker port <container ID>；
* 查看容器内应用的日志：docker log <container ID>；
* 容器内部运行的进程：docker top <container name>；
* 查看容器的底层信息：docker inspect <container name>；
* 退出当前容器但其继续运行：Ctrl + Q + P；

### 镜像

* 查看本地主机上的镜像：docker images；
* 拉取一个镜像：docker pull <images>:<edition>；
* 查找位于hub的镜像信息：docker search <images>；
* 删除一个镜像：docker rmi <images>；